POS Tagger and Chunker for Tamil Language

(தமிழ் சொல்வகை அடையாளப்படுத்தி மற்றும் தொடர் பகுப்பான்)

Dhanalakshmi V[1], Anand kumar M[1], Rajendran S[2], Soman K P[1]
{v_dhanalakshmi, m_anandkumar, kp_soman} @ettimadai.amrita.edu, raj_ushus@yahoo.com

[1]Amrita Vishwa Vidyapeetham, Ettimadai, Coimbatore, Tamilnadu, India.
[2]Tamil University, Thanjavur, Tamilnadu, India.

Abstract

This paper presents the Part Of Speech tagger and Chunker for Tamil using Machine learning techniques. Part Of Speech tagging and chunking are the fundamental processing steps for any language processing task. Part of speech (POS) tagging is the process of labeling automatic annotation of syntactic categories for each word in a corpus. Chunking is the task of identifying and segmenting the text into syntactically correlated word groups. These are done by the machine learning techniques, where the linguistical knowledge is automatically extracted from the annotated corpus. We have developed our own tagset for annotating the corpus, which is used for training and testing the POS tagger generator and the chunker. The present tagset consists of thirty-two tags for POS and nine tags for chunking. A corpus size of two hundred and twenty five thousand words was used for training and testing the accuracy of the POS tagger and Chunker. We found that SVM based machine learning tool affords the most encouraging result for Tamil POS tagger (95.64%) and chunker (95.82%).

1 Introduction

Part of speech (POS) tagging and chunking are well studied problems in the field of Natural Language Processing (NLP). Different approaches have already been tried to automate the task of POS tagging and chunking for English and other languages. The basic processing step consists of assigning POS tags to every token in the text. A subsequent step after POS tagging focuses on the identification of basic structural relations between groups of words in a sentence. This recognition is usually referred to as chunking. It is essential for many NLP tasks such as structure identification, information extraction, parsing and phrase based machine translation system. Chunker divides a sentence into its major-non-overlapping phrases and attaches a label to each chunk. Chunking falls between tagging and parsing. The structure of individual chunks is fairly easy to describe, while relations between chunks are harder and more dependent on individual lexical properties.

The capability for a computer to automatically POS tag and chunk a sentence is very essential for further analysis in many approaches to the field of NLP. Many of the machine learning techniques and algorithms are used in this task. Our POS tagger and chunker based on machine learning techniques using SVM are trained and tested with the tagged corpus of size about two lakh and twenty five thousand words.

2 POS Tagging in Tamil

The Part of speech (POS) tagging is the process of labeling a part of speech or other lexical class marker to each and every word in a sentence. It is similar to the process of tokenization for computer languages. POS tagging is considered as an important process in speech recognition, natural language parsing, information retrieval and machine translation. Tamil being a Dravidian language has a very rich morphological structure which is agglutinative. Tamil words are made up of lexical roots followed by one or more affixes. So tagging a word in a language like Tamil is very complex. The main challenges in Tamil POS tagging are solving the complexity and ambiguity of words [Dhanalakshmi V et al., 2009].

Various methodologies have been developed for POS Tagging in different languages. In case of Tamil language a rule-based POS tagger for Tamil was developed and tested [Arulmozhi et al., 2004]. This system gives only the major tags and the sub tags are overlooked while evaluation. A hybrid POS tagger for Tamil using HMM technique and a rule based system was also developed [Arulmozhi P and Sobha L, 2006].

Our POS tagger is based on machine learning techniques using SVM. We tagged our raw corpus of size about two hundred and twenty five thousand words using our Amrita tag set and then trained our corpus with the machine learning based SVMTool by tuning the parameters and feature patterns based on Tamil language. A raw corpus was tested using SVMTool and obtained an overall accuracy of 95.64%.

3 Customized POS Tagset

Many tagsets are already in existence for Tamil (AUKBC, Vasuranganathan tagset, CIIL Tagset for Tamil**,** etc). However, we encountered the following problems with these tagsets:

1. For each word, the grammatical categories as well as grammatical features are considered. Hence we need to split each and every inflected word in the corpus, which makes the tagging process very complex.
2. The number of tags is very large. This leads to increased complexity during POS tagging which in turn reduces the tagging accuracy.

For simple POS level, we wanted a tagset which has just the grammatical categories excluding grammatical features. Since the grammatical features can be obtained from the morphological analyzer. We needed a tagset with minimum tags without compromising on tagging efficiency. Hence we decided to create our own tagset for Tamil following the guidelines as mentioned in AnnCorra, Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages [Akshar Bharati et al., 2006].

Our customized tagset uses only 32 tags. We do not consider the inflections or the grammatical features of the words. We use compound tag for compound nouns (NNC) and compound proper nouns (NNPC). We consider the tag VBG for verbal nouns and participle nouns. The tagset is shown in the figure below:

S.No	*POS*	*Description*		*S.no*	*POS*	*Description*
1	<NN>	NOUN		17	<VINT>	VERB INFINITE
2	<NNC>	COMPOUND NOUN		18	<CNJ>	CONJUNCTION
3	<NNP>	PROPER NOUN		19	<CVB>	CONDITIONAL VERB
4	<NNPC>	COMPOUND PROPER NOUN		20	<QW>	QUESTION WORD
5	<ORD>	ORDINALS		21	<COM>	COMPLEMENTIZER
6	<CRD>	CARDINALS		22	<NNQ>	QUANTITY NOUN
7	<PRP>	PRONOUN		23	<QTF>	QUANTIFIERS
8	<PRIN>	PRONOUN INTROGATIVE		24	<PPO>	POSTPOSITIONS
9	<PRID>	PRONOUN INDEFINITE		25	<DET>	DETERMINERS
10	<ADJ>	ADJECTIVE		26	<INT>	INTENSIFIER
11	<ADV>	ADVERB		27	<ECH>	ECHO WORDS
12	<VNAJ>	VERB NON FINITE ADJECTIVE		28	<EMP>	EMPHASIS
13	<VNAV>	VERB NON FINITE ADVERB		29	<COMM>	COMMA
14	<VBG>	VERBAL GERUND		30	<DOT>	DOT
15	<VF>	VERB FINITE		31	<QM>	QUESTION MARKS
16	<VAX>	VERB AUXILARY		32	<RDW>	REDUPLICATION WORDS

Figure 1. Amrita POS Tagset

4 Chunking in Tamil

A typical chunk consists of a single content word surrounded by a constellation of function words [S.Abney, 1991]. Chunks are normally taken to be a non recursive correlated group of words. Tamil being an agglutinative language have a complex morphological and syntactical structure. It is a relatively free word order language but in the phrasal and clausal construction

it behaves like a fixed word order language. So the process of chunking in Tamil is less complex compared to the process of POS tagging. Various methodologies have been developed for chunking in different languages. In Tamil language TBL was used for text chunking [Sobha L et al., 2006]. vaanavil of RCILTS identifies the syntactic constituents of a Tamil sentence. Our Chunker is based on machine learning techniques (YamCha) using SVM.

4.1 Customized Chunk Tagset

We followed the guidelines mentioned in AnnCorra, while creating our tagset for chunking. Our Amrita chunking tagset contains nine tags. The tagset is described below:

Noun Chunks will be given the tag NP. It includes non-recursive noun phrases and postpositional phrases. The head of a noun chunk would be a noun. Noun qualifiers like adjective, quantifiers, determiners will form the left side boundary for a noun chunk and the head noun will mark the right side boundary for it. Examples for NP chunk are given below.

[அந்த <DET> (B-NP) அழகான <ADJ> (I-NP) பெண் <NN> (I-NP)] NP

An adjectival chunk is tagged as AJP. This chunk will consist of all adjectival chunks including the predicative adjectives. However, adjectives appearing before a noun will be grouped together with the noun chunk.

[திரைப்படம் <NN> (B-AJP) சார்ந்த <ADJ> (I-AJP)] AJP

Adverbial chunk <AVP> is tagged accordance with the tags used for POS tagging.

[அருகே <ADV> (B-AVP)]AVP

Conjunctions are the words used to join individual words, phrases, and independent clauses. It is labeled as CJP.

[ஆனால் <CNJ>(B-CJP)] CJP

Complimentizer are the words equivalent to the term subordinating conjunction in traditional grammer. For example, the word *that* is generally called a Complimentizer in English. In Tamil, *enru* and its variations falls into this category. Complimentizer is tagged in accordance with the tages used for POS tagging. It is tagged as COMP.

[என்று <COM> (B-COMP)] COMP

Verb chunks are mainly classified into Verb finite chunk and verb non-finite chunk. Verb finite chunk includes main verb and its auxiliaries. It is tagged as VFP. Examples for verb – finite chunk are given below.

[உள்ளது<VF> (B-VFP)] VFP

Non-finite verb comprise all the non-finite form of verbs. In Tamil we have four non-finite forms i.e., relative participle, adverbial participle, conditional and infinitive verb. It is tagged as VNP. Examples for verb non-finite chunk are given below.

[வெளிவந்த **(VNAJ) (B-VNP)] VNP** செய்திக் <NNC> < B-NP> குறிப்பு <I –NP> <NNC>

[விரைந்து **<VNAV>(B-VNP)] VNP** முடித்தான் <VF>

Gerundial forms are repersented by a seperate chunk. It is tagged as VGP. Example for gerundial chunk is given below.

தொழிற்சாலை <NN> **[**அமைப்பதில் **<VBG>(B-VGP)] VGP** தாமதம் <NN>

Symbols like .(Dot) and ? (question mark) are tagged as <**O**> . , (Comma) is tagged with the preceeding tag.

5 Corpus Development

POS tagged corpus containing two lakh and twenty five thousand words was prepared by collecting corpora from Dinamani newspaper, yahoo Tamil news, online Tamil short stories, etc Dhanalakshmi.V et al., 2008. This POS tagged corpus is used for chunking corpus development. Our customized tagset was used to tag the POS tagging and chunking corpus. The tagged corpus is given for training using the machine learning tools. After training, the untagged corpus is tagged by tagger generator. The output of tagger generator is manually corrected to increase the corpus size.

Training data format*:* The training data should be in a particular format. The training data must consist of multiple tokens, these token are nothing but words, and a sequence of token becomes a sentence. Each token should be represented in one line, with the columns separated by white space. Many numbers of columns can be used, but the columns are fixed through all tokens. There should be some kinds of 'semantics' among the columns, i.e. first column is a 'word', second column is 'pos tag', and third column is 'chunk tag' and so on. The last column represents the answer tag which is going to be trained by SVM based Tools. We have fixed three column formats. Following is a sample of the training data.

வளாகத் <NNC> <B-NP>

தேர்வில் <NNC> <I-NP>

வேலைவாய்ப்பு <NN> <B-NP>

பெற்ற <VNAJ> <B-VNP>

மாணவர்களின் <NN> <B-NP>

பட்டியல் <NN> <I-NP>

வெளியிடும் <VNAJ> <B-VNP>

விழா <NN> <B-NP>

திங்கள்கிழமை <NNP> <B-NP>

நடைபெற்றது <VF> <B-VFP>

. <DOT> <O>

6 SVM based Tools for Tamil POS Tagger and chunker

The SVMTool is a simple, flexible, and effective generator of sequential taggers based on Support Vector Machines and how it is being applied to the problem of part-of-speech tagging. This SVM-based tagger is robust and flexible for feature modeling (including lexicalization), trains efficiently, and is able to tag thousands of words per second. YamCha(Yet Another Multipurpose Chunk Annotator by Taku Kudo) is a generic, customizable, and open source text chunker. Yamcha is using a state-of-the-art machine learning algorithm called Support Vector Machines (SVMs), introduced by Vapnik.

6.1 Support Vector Machine

SVM is a machine learning algorithm for binary classification, which has been successfully applied to a number of practical problems, including NLP. Tagging a word in context is a multi-class classification problem. Since SVMs in general are binary classifiers, a binarization of the problem must be performed initially before applying them. Here a simple one-per-

class binarization is applied, i.e., a SVM is trained for every POS tag in order to distinguish between examples of this class and all the rest. When tagging a word, the most possible tag according to the predictions of all binary SVMs is selected.

6.2 SVMTool for Tamil POS Tagger

The SVMTool software package consists of three main components, namely the model learner (SVMTlearn), the tagger (SVMTagger) and the evaluator (SVMTeval).

SVM model is learned from a training corpus using the SVMTlearn component. Different models are learned for the different tagging strategies. During tagging time, the SVMTagger component is used to choose the tagging strategy that is most suitable for the purpose of the tagging. Finally, when we give a correctly tagged corpus and the corresponding SVMTool predicted annotation, the SVMTeval component displays tagging results and reports. Tagged corpus is used for training a set of SVM classifiers. This is done using SVMlight, an implementation of Vapnik's SVMs in C, developed by Thorsten Joachims.

6.3 Yamcha for Tamil Chunker

YamCha is an open source text chunker and so called Support Vector machines (SVMs). SVMs are binary classifiers and thus must be extended to multiclass classifiers to classify three cases for NP chunking with (I, O, B). By mapping the n-dimensional input space into high dimensional feature space in which a linear classifier is then typically constructed. This approach is used for chunking, YamCha is used to perform the initial tagging, basic features in Yamcha are used, later all possible POS tag for the words in the corpus are added. This information is added to the training corpus and then it is trained using SVM thereby predicting the chunk boundary names using Yamcha, Finally the chunk labels and the chunk boundary names are merged to obtain the chunk tag.

7 Conclusion

This paper has described the POS tagger and Chunker for Tamil using Machine learning approach. For the POS tagging and chunking we have used a corpus of size 2, 25,000 words. The corpus is divided into training set (1, 65,000 words) and test set (60,000 words). Machine learning tools like SVMTool and Yamcha are trained and tested for the same corpus. We have found that automatic POS tagging and chunking done by SVM based Machine learning tools gives better result. A GUI to enhance the user friendliness of the tool was also developed.

References

Akshar Bharati, Rajeev Sangal, Dipti Misra Sharma and Lakshmi Bai. 2006. AnnCorra:*Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages*, Technical Report, Language Technologies Research Centre IIIT, Hyderabad.

Arulmozhi P, Sobha L, 2006. *A Hybrid POS Tagger for a Relatively Free Word Order Language.* In proceedings of MSPIL-2006, Indian Institute of Technology, Bombay.

Dhanalakshmi V, Anandkumar M, Shivapratap G, Soman, K P, Rajendran S. May 2009. *Tamil POS Tagging using Linear Programming,* In International Journal of Recent Trends in Engineering, 1(2):166-169.

Gim´enez, J and L M`arquez, 2003. *Fast and Accurate Part of- Speech Tagging: The SVM Approach* Revisited, in Proceedings of the Fourth RANLP.

Sobha L,Vijay Sundar Ram R. 2006. *Noun Phrase Chunking in Tamil*, In proceeding of the MSPIL-06, Indian Institute of Technology, Bombay.pp-194-198.

Taku Kudo, Yuji Matsumoto. 2001.YamCha: *Yet Another Multipurpose Chunk Annotator* http://chasen.org/~taku/software/YamCha/.